

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів  
Кафедра систем управління літальних апаратів

## Лабораторна робота № 4

з дисципліни «Алгоритмізація та програмування»  
на тему "Структурування програм з використанням функцій"

ХАІ.301. 174. 319. 5ЛР

Виконав студент гр. 319

Женя Качка

(підпис, дата)

(П.І.Б.)

Перевірив

\_\_\_\_\_ К.Т.Н.,

доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

2024

## МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису оголошення, визначення і виклику функцій в C++ і реалізувати консольний додаток з використанням функцій з параметрами і поверненням результату на мові програмування C++ в середовищі Visual Studio.

## ПОСТАНОВКА ЗАДАЧІ

**Proc27** Описати функцію `RectP (x1, y1, x2, y2)` дійсного типу, яка обчислює периметр прямокутника зі сторонами, паралельними осям координат, за дійсними координатами  $(x1, y1)$ ,  $(x2, y2)$  його протилежних вершин. За допомогою цієї функції знайти периметри трьох прямокутників з даними протилежними вершинами.

**Boolean20** Дано тризначне число. Перевірити істинність висловлювання: «Все цифри даного числа різні».

**Integer30** Дано номер деякого року (ціле додатне число). Визначити відповідний йому номер століття, враховуючи, що, наприклад, початком 20 століття був 1901 рік.

## ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі **Proc27**

Вхідні дані (ім'я, опис, тип, обмеження):

$x1, y1, x2, y2$  , координати протилежних вершин прямокутника , дійсні

Вихідні дані (ім'я, опис, тип):

$P$  , периметр прямокутника , дійсне

Алгоритм вирішення показано на малюнку .1

малюнок 1.

Малюнок 1 – задача Proc27

```
#include <iostream>
#include <cmath> // Для використання std::abs
//перша програма

// Функція RectP для обчислення периметра прямокутника
double RectP(double x1, double y1, double x2, double y2) {
    // Знаходимо довжини сторін прямокутника
    double width = std::abs(x2 - x1); // Ширина прямокутника
    double height = std::abs(y2 - y1); // Висота прямокутника

    // Периметр прямокутника: 2 * (ширина + висота)
    return 2 * (width + height);
}

int main() {
    setlocale(LC_ALL, "Ru");
    // Координати трьох прямокутників
    double x1_1 = 1, y1_1 = 2, x2_1 = 4, y2_1 = 6; // Перший прямокутник
    double x1_2 = -3, y1_2 = -1, x2_2 = 2, y2_2 = 3; // Другий прямокутник
    double x1_3 = 0, y1_3 = 0, x2_3 = 5, y2_3 = 7; // Третій прямокутник

    // Знаходимо периметри прямокутників
    double perimeter1 = RectP(x1_1, y1_1, x2_1, y2_1);
    double perimeter2 = RectP(x1_2, y1_2, x2_2, y2_2);
    double perimeter3 = RectP(x1_3, y1_3, x2_3, y2_3);

    // Виводимо результати
    std::cout << "Периметр першого прямокутника: " << perimeter1 << std::endl;
    std::cout << "Периметр другого прямокутника: " << perimeter2 << std::endl;
    std::cout << "Периметр третього прямокутника: " << perimeter3 << std::endl;

    //друга програма
}
```

Лістинг коду вирішення задач Proc27, Boolean20, Integer30.

наведено в дод. А (стор. 7).

Екран роботи програми показаний на рис. Б.5.

Завдання 2.

Вирішення задачі Boolean20

Вхідні дані (ім'я, опис, тип, обмеження):

N , Тризначне число , ціле ,  $100 \leq N \leq 999$

Вихідні дані (ім'я, опис, тип):

result , Істинність висловлювання , Логічний

Алгоритм вирішення показано на малюнку .2

## Малюнок 2 – задача Boolean20

```
//друга програма

int number;
setlocale(LC_ALL, "Ru");

// Введення тризначного числа
std::cout << "Введіть тризначне число: ";
std::cin >> number;

// Перевірка, чи є число тризначним
if (number < 100 || number > 999) {
    std::cout << "Число повинно бути тризначним!" << std::endl;
    return 1;
}

// Розбиття числа на цифри
int digit1 = number / 100;           // Сотні
int digit2 = (number / 10) % 10;     // Десятки
int digit3 = number % 10;           // Одиниці

// Перевірка, чи всі цифри різні
if (digit1 != digit2 && digit1 != digit3 && digit2 != digit3) {
    std::cout << "Істинно: всі цифри різні." << std::endl;
}
else {
    std::cout << "Хибно: є однакові цифри." << std::endl;
}
```

Завдання 3.

Вирішення задачі Integer30.

Вхідні дані (ім'я, опис, тип, обмеження):

year , Номер року , ціле , рік>0

Вихідні дані (ім'я, опис, тип):

century , Номер століття , ціле

Алгоритм вирішення показано на малюнку .3

### Малюнок 3 – задача Integer30.

```
// третья программа
int main();
setlocale(LC_ALL, "Ru");
int year;

// Введення року
std::cout << "Введіть номер року (ціле додатне число): ";
std::cin >> year;

// Перевірка на коректність введення
if (year <= 0) {
    std::cout << "Рік повинен бути додатним числом!" << std::endl;
    return 1;
}

// Визначення століття
int century = (year - 1) / 100 + 1;

// Виведення результату
std::cout << "Рік " << year << " відповідає " << century << " століттю." << std::endl;

return 0;
}
```

## ВИСНОВКИ

Лабораторна робота була спрямована на практичну реалізацію функцій у мові C++ для розв'язання трьох завдань:

1. Обчислення периметра прямокутника за координатами його вершин за допомогою функції `RectP(x1, y1, x2, y2)`. У роботі реалізовано програму, яка дозволяє вводити координати трьох прямокутників і обчислювати їх периметри.
2. Перевірка унікальності цифр тризначного числа. Реалізовано функцію `AllDigitsDifferent(int n)`, яка визначає, чи всі цифри введеного числа є різними.
3. Визначення століття за роком. Реалізовано функцію `CenturyFromYear(int year)`, яка на основі введеного року обчислює номер відповідного століття.

## ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
#include <cmath> // Для використання std::abs
//перша програма
// Функція RectP для обчислення периметра прямокутника
double RectP(double x1, double y1, double x2, double y2) {
    // Знаходимо довжини сторін прямокутника
    double width = std::abs(x2 - x1); // Ширина прямокутника
    double height = std::abs(y2 - y1); // Висота прямокутника
    // Периметр прямокутника: 2 * (ширина + висота)
    return 2 * (width + height);
}

int main() {
    setlocale(LC_ALL, "Ru");
    // Координати трьох прямокутників
    double x1_1 = 1, y1_1 = 2, x2_1 = 4, y2_1 = 6; // Перший прямокутник
    double x1_2 = -3, y1_2 = -1, x2_2 = 2, y2_2 = 3; // Другий прямокутник
    double x1_3 = 0, y1_3 = 0, x2_3 = 5, y2_3 = 7; // Третій прямокутник
    // Знаходимо периметри прямокутників
    double perimeter1 = RectP(x1_1, y1_1, x2_1, y2_1);
    double perimeter2 = RectP(x1_2, y1_2, x2_2, y2_2);
    double perimeter3 = RectP(x1_3, y1_3, x2_3, y2_3);
    // Виводимо результати
    std::cout << "Периметр першого прямокутника: " << perimeter1 <<
std::endl;
    std::cout << "Периметр другого прямокутника: " << perimeter2 <<
std::endl;
    std::cout << "Периметр третього прямокутника: " << perimeter3 <<
std::endl;
    //друга програма
    int number;
    setlocale(LC_ALL, "Ru");
    // Введення тризначного числа
    std::cout << "Введіть тризначне число: ";
    std::cin >> number;
    // Перевірка, чи є число тризначним
    if (number < 100 || number > 999) {
        std::cout << "Число повинно бути тризначним!" << std::endl;
        return 1;
    }
    // Розбиття числа на цифри
    int digit1 = number / 100; // Сотні
    int digit2 = (number / 10) % 10; // Десятки
    int digit3 = number % 10; // Одиниці
    // Перевірка, чи всі цифри різні
    if (digit1 != digit2 && digit1 != digit3 && digit2 != digit3) {
        std::cout << "Істинно: всі цифри різні." << std::endl;
    }
    else {
        std::cout << "Хибно: є однакові цифри." << std::endl;
    }
    return 0;
}

//третья програма
int main();
setlocale(LC_ALL, "Ru");
int year;
// Введення року
std::cout << "Введіть номер року (ціле додатне число): ";
std::cin >> year;
```

```

#include <iostream>
#include <cmath> // Для використання std::abs
//перша програма
// Функція RectP для обчислення периметра прямокутника
double RectP(double x1, double y1, double x2, double y2) {
    // Знаходимо довжини сторін прямокутника
    double width = std::abs(x2 - x1); // Ширина прямокутника
    double height = std::abs(y2 - y1); // Висота прямокутника
    // Периметр прямокутника: 2 * (ширина + висота)
    return 2 * (width + height);
}

int main() {
    setlocale(LC_ALL, "Ru");
    // Координати трьох прямокутників
    double x1_1 = 1, y1_1 = 2, x2_1 = 4, y2_1 = 6; // Перший
прямокутник
    double x1_2 = -3, y1_2 = -1, x2_2 = 2, y2_2 = 3; // Другий
прямокутник
    double x1_3 = 0, y1_3 = 0, x2_3 = 5, y2_3 = 7; // Третій
прямокутник
    // Знаходимо периметри прямокутників
    double perimeter1 = RectP(x1_1, y1_1, x2_1, y2_1);
    double perimeter2 = RectP(x1_2, y1_2, x2_2, y2_2);
    double perimeter3 = RectP(x1_3, y1_3, x2_3, y2_3);
    // Виводимо результати
    std::cout << "Периметр першого прямокутника: " << perimeter1 <<
std::endl;
    std::cout << "Периметр другого прямокутника: " << perimeter2 <<
std::endl;
    std::cout << "Периметр третього прямокутника: " << perimeter3 <<
std::endl;
    //друга програма
    int number;
    setlocale(LC_ALL, "Ru");
    // Введення тризначного числа
    std::cout << "Введіть тризначне число: ";
    std::cin >> number;
    // Перевірка, чи є число тризначним
    if (number < 100 || number > 999) {
        std::cout << "Число повинно бути тризначним!" << std::endl;
        return 1;
    }
    // Розбиття числа на цифри
    int digit1 = number / 100; // Сотні
    int digit2 = (number / 10) % 10; // Десятки
    int digit3 = number % 10; // Одиниці
    // Перевірка, чи всі цифри різні
    if (digit1 != digit2 && digit1 != digit3 && digit2 != digit3) {
        std::cout << "Істинно: всі цифри різні." << std::endl;
    }
    else {
        std::cout << "Хибно: є однакові цифри." << std::endl;
    }
    return 0;
}

//третья програма
int main();
setlocale(LC_ALL, "Ru");
int year;
// Введення року
std::cout << "Введіть номер року (ціле додатне число): ";
std::cin >> year;

```



```
// Перевірка на коректність введення
if (year <= 0) {
    std::cout << "Рік повинен бути додатним числом!" << std::endl;
    return 1;
}
// Визначення століття
int century = (year - 1) / 100 + 1;
// Виведення результату
std::cout << "Рік " << year << " відповідає " << century << "
століттю." << std::endl;
return 0;
}
```

## ДОДАТОК Б

### Скрін-шоти вікна виконання програми

```
Периметр першого прямокутника: 14
Периметр другого прямокутника: 18
Периметр третього прямокутника: 24

C:\Users\tvink\source\repos\практическая номер 4\x64\Debug\практическая номер 4.exe (процесс 14824) завершил работу с
дом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання  
Proc27

```
Введите трёхзначное число: 123
Все цифры разные.

C:\Users\tvink\source\repos\практическая номер 4\x64\Debug\практическая номер 4.exe (процесс 4452) завершил работу
ом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка"
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання  
Boolean20

```
Введите год: 2024
Век: 21

C:\Users\tvink\source\repos\практическая номер 4\x64\Debug\практическая номер 4.exe (процесс 7460) завершил работу
ом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка"
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок Б.3 – Экран виконання програми для вирішення завдання Integer30.